Article

Cross Entropy Covariance Matrix Adaptation Evolution Strategy solving the Bi-level Bidding Optimization Problem in Local Energy Markets with Renewables

Dharmesh Dabhi[1], Kartik Pandya[1], Joao Soares[2*], Fernando Lezama[2] and Zita Vale[2]

[1] M & V Patel Dept. of Electrical Engineering, CSPIT, CHARUSAT UNIVERSITY, CHANGA-388421, India; dharmeshdabhi.ee@charusat.ac.in (D.D.); kartikpandya.ee@charusat.ac.in (K.P.)
[2] GECAD, School of Engineering, Polytechnic of Porto, Porto 4200-072, Portugal; flz@isep.ipp.pt (F.L.); zav@isep.ipp.pt (Z.V.)
* Correspondence: jan@isep.ipp.pt (J.S.);

Abstract: The increased penetration of renewables in distribution power systems has motivated researchers to take significant interest in local energy transactions. The major goal of Local Energy Markets (LEM) is to promote the participation of small consumers in energy transactions and providing an opportunity for transactive energy systems. Such energy transactions in LEM are considered as a bi-level optimization problem in which all agents at upper and lower levels try to maximize their profits. But typical bi-level problem is very complex as it is inherently nonlinear, discontinued and strongly NP-hard. So, this article proposes the application of hybridized Cross Entropy Covariance Matrix Adaptation Evolution Strategy (CE-CMAES) to tackle such a complex bi-level problem of LEM. The proposed CE-CMAES secured the 1st rank in Testbed-2 entitled, “Bi-level optimization of end-users’ bidding strategies in local energy markets (LM)” at international competitions on Smart Grid Problems, held at GECCO 2020 and WCCI 2020. CE method is used for global exploration of search space and CMAES is used for local exploitation as its adaptive step-size mechanism prevents its premature convergence. A practical distribution system with renewable energy penetration is considered for simulation. The comparative analysis shows that the overall cost, mean fitness and Ranking Index (R.I) obtained from CE-CMAES are superior to those obtained from the state-of-the-art participated algorithms. Wilcoxon Signed Rank Statistical test also proves that CE-CMAES is statistically different from the tested algorithms.

Keywords: Bi-level Problem; Covariance Matrix; Cross-Entropy Method; Local Energy Market; Optimal Bidding.

1. Introduction

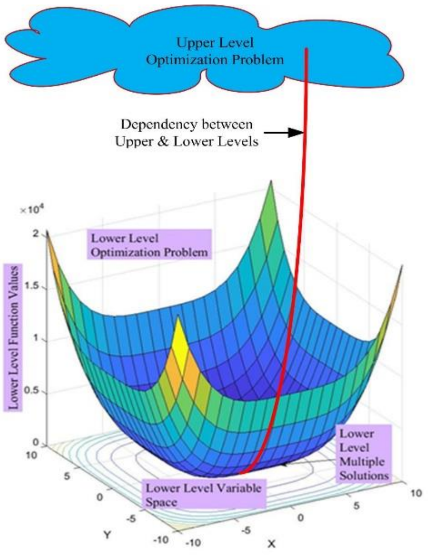
The smart grid era has created lots of new challenges and opportunities that must be explored and implemented to optimize its performance [1]. The large penetration of distributed renewable energy sources has created significant opportunity in energy transactions at the local lower level [2]. In this paradigm, local energy markets (LEM) facilitates small agents to trade renewable energy at the local level and contribute to the reduction of environmental pollutants. Thus, LEM empowers them to take a large part in the energy community to implement fully transactive energy systems [3]. The energy transactions in LEM results in to a bi-level optimization problem in which all agents try to maximize their profits by optimizing their bidding strategies. In this regard, [4] proposed bi-level problem to optimize offering prices of energy producers and subsequently it was converted into a mixed integer linear programming. But demand side bidding was ignored and not optimized.Proposed optimal bidding strategy of transactive agents in which two subproblems were considered to reduce no. of variables and nonlinear constraints were replaced by linear inequalities to reduce the computational burden [5]. But the effect of renewables was ignored in the problem formulation, which is important for modern distribution system. Authors of [6] proposed a two-stage robust model to optimize bidding in day-ahead and real time markets in virtual power plant environment. The optimization problem was formulated using linear programming. But since the optimization problem was linearized, the quality of the optimal solutions was also compromised. Authors of [7] proposed hybrid optimal bidding strategy for a microgrid in day-ahead market under the influence of intermittent distributed generation and price responsive loads. Stochastic models were used to model day-ahead market price and uncertain output of intermittent DG while MILP was used to limit power imbalance in real time market considering uncertainty in real time market price. The hourly bidding curves obtained by such approach were less accurate as the original nonlinear problem was considered as a linear. [8] Proposed a robust based bidding and offering strategy for a price-maker energy storage facility participating in a day-ahead market. It proposed max-min MILP approach to present a participation strategy to manage the risk of uncertainty associated with forecasted hourly generation and demand price quota curves. Some authors have explored metaheuristic optimization methods to solve bi-level problem in LEM. An example of such application can be found in [9], where Particle Swarm Optimization (PSO) was used to solve bi-level problem in which power producers' biddings were optimized at upper level in monthly contract and balancing markets. In this, a basic PSO is implemented whose local exploitation search ability is poor and thus suffers with pre-mature convergence. Authors of [10] used basic Genetic Algorithm (GA) to optimize the bidding of price maker retailers with probability-based estimation of market price. But the GA is quite computationally expensive. In [11] Ant colony optimization was proposed to optimize biddings of all market participants in LEM.

The literature survey reveals that the bidding optimization problems of the market participants are generally formulated as a bi-level optimization problem. In the majority of the surveyed papers, mixed integer linear programming method was used. In this, the objective functions/constraints are linearized and as a result the accuracy of the optimal solutions is decreased. But typically, any bi-level optimization problem is inherently highly nonlinear [12], non-convex, non-differentiable, discontinued and strongly NP-hard [13], because the lower-level problem has nonlinear inducible region and it may have multiple optimal solutions as shown in Figure. 1. Also, upper and lower level problems are mutually dependent. Remaining surveyed papers have used basic meta-heuristic methods to solve bi-level problems. Such methods are either vulnerable to pre-mature convergence or computationally expensive.

So, to get rid of aforementioned problems related to bi-level optimization, this article provides the following contributions to the available literature.

- A multi-period bi-level optimization problem is formulated in which competitive agents in the upper-level problem try to maximize their profits, modifying and depending on the price determined in lower-level problem (i.e. the clearing price in the LEM), hence resulting in a strong interdependence of their decisions.
- The applications of 1st rank winner optimization algorithm entitled "CE-CMAES" in Test Bed 2: "bi-level optimization of end-users' bidding strategies in local energy markets" at international competitions entitled, "Evolutionary Computation in Uncertain Environments: A Smart Grid Application" organized at Genetic and Evolutionary Computation Conference (GECCO 2020) and IEEE world Congress on Computational Intelligence (WCCI 2020) have been proposed to solve bi-level optimal bidding in local energy markets with renewable energy sources.
- Practical distribution power system with renewable energy sources has been used to test the effectiveness of the proposed algorithm.
- The comparative analysis shows that the profits of all agents obtained from CE-CMAES are better than those obtained from the state-of-the-art algorithms in a day-ahead local energy market.

The remaining paper is organized in six sections. Following this introduction, the formulation of bi-level optimization problem is discussed in section 2. Section 3 discusses the CE-CMAES algorithm in detail. In section 4, the comparative analysis of the proposed method with the state-of-the-art optimization methods is carried out on a practical distribution system. Eventually, section 5 outlines main conclusions and prospective future research work. In section 6 references are given.



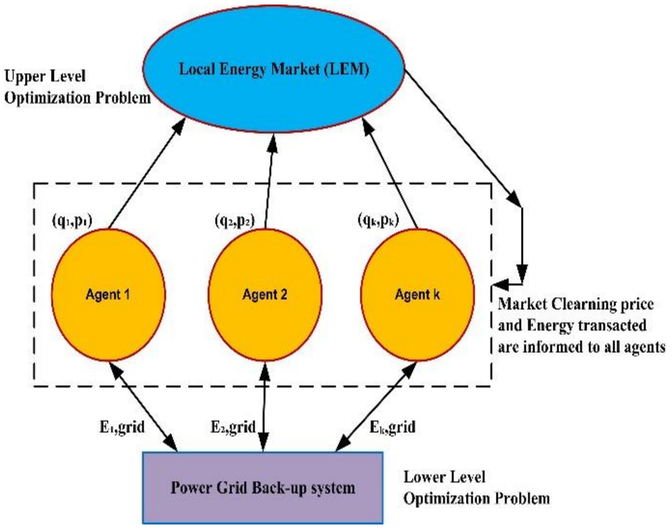

Upper Level Optimization Problem
Local Energy Market (LEM)
(q1,p1)
(q2,p2)
(qk,pk)
Agent 1
Agent 2
Agent k
Market Clearning price and Energy transacted are informed to all agents
E1,grid
E2,grid
Ek,grid
Power Grid Back-up system
Lower Level Optimization Problem

Figure 1. Graphical Representation of Bi-level Optimization Problem

Figure 2. LEM and the main grid as back-up system [14]

2. Optimization Problem Formulation

A day-ahead Local Energy Market (LEM) bidding optimization problem is proposed in [14] in which agents submit offers and bids to maximize their profits and minimize their costs respectively. The agents considered in LEM are of the 3 types, namely prosumers (consumers with power generation capabilities), small producers and consumers. Also, all agents have access to the main grid, which works as a back-up power system. So, all agents can trade energy in LEM with prices between the feed-in tariff $\left(c^F\right)$ and Grid electricity tariff $\left(c^G\right)$. It is assumed that $c^F < c^G$. So, it is assumed that direct energy transactions with the main grid are less beneficial to the agents than in the LEM. Figure. 2 shows the LEM scenario.

2.1 Bi-level optimization problem formulation

The LEM bidding optimization problem is basically a bi-level optimization problem in which the upper-level problem and lower-level problem correspond to the maximization of agents' profits and maximization of energy transacted respectively in LEM [14]. So, the solution of the lower-level affects the upper-level by modifying the profits of all agents.

2.1.1 Upper Level Problem

Assume a set of producer agents $J = 1, 2, ..., N_p$ and a set of consumer agents $I = 1, 2, ..., N_c$ where each agent j wants to maximize its profit and each agent i wants to minimize its costs.

The optimization problem for all producer agents can be formulated as Equation (1) in which they try to maximize their profits regarding their production marginal cost.

$$max.\, P_j = \sum_i cp\ x_{j,i} + c^F\ Esell_{j,grid} - c^{mc} G_j \tag{1}$$

Where, P_j is the profit of agent j, cp is the LEM clearing price (equal for buyers and sellers), $x_{j,i}$ is the energy sold by agent j to agent i, c^F is the feed-in tariff, $Esell_{j,grid}$ is the energy sold by agent j to grid, and $c^{mc} G_j$ is the marginal cost associated to j. c^{mc} is taken as zero for PV generation. Let, $c^{mc} = c^{mc}_{CHP}\left(G_j\right)$ is the marginal cost associated to a Combined Heat and Power (CHP) generator and it is defined as a monotonically decreasing function [15] as given in Equation (2).

$$c^{mc}_{CHP}\left(G_j\right) = \frac{b_{CHP}\sqrt{G_j}}{G_j} \tag{2}$$

Where, b_{CHP} is a constant cost factor of the CHP generation unit and G_j is the energy produced by the CHP unit.

The optimization problem for all consumer agents can be formulated as Equation (3) in which they try to minimize their costs.

$$min.\, C_i = \sum_j cp\ x_{j,i} + c^G\ Ebuy_{i,grid} \tag{3}$$

where C_i is the cost of agent i, cp is the LEM clearing price, $x_{j,i}$ is the energy sold by agent j to agent i in the LEM, c^G is the Grid price, and $Ebuy_{i,grid}$ is the energy bought by agent i from the grid.

2.1.2 Lower Level Problem

It is evident that the agents' profits and costs are dependent upon the LEM clearing price, which is determined in the lower-level problem and depends upon the bidding process. The lower-level problem is modelled as an asymmetric pool market in which offers and bids are allocated using a merit order mechanism to determine the supply and demand curves for energy [6]. It determines the clearing price (cp) and energy transacted between the agents. A simple deterministic mechanism is used to determine the amount of energy traded to the grid (i.e., variables $Esell_{j,grid}$ and $Ebuy_{i,grid}$ in Equation (1) and Equation (3) respectively.) Producer j determines the energy sold to the grid as per Equation (4) regarding its marginal cost c^m.

$$Esell_{j,grid} = \begin{Bmatrix} P_{max,j} - \sum_i x_{j,i}\ \ if\ \ P_{marginal,j} > P_j \\ O \qquad\qquad Otherwise \end{Bmatrix} \tag{4}$$

Where, $P_{marginal,j}$ are the profits that agent j could make considering the marginal cost of producing its maximum capacity and selling the non-traded energy in the LEM to the grid. P_j are the regular profits that the agent would make by selling only into the LEM. As a result, it is guaranteed that producers only sell energy to the grid if they can actually get some profits with their remaining capacity. In case of PV generation where the marginal cost is almost zero, the agents will always sell its remaining energy to the grid as per this rule. Consumer i decides the energy bought from the grid as per Equation (5).

$$Ebuy_{i,grid} = L_{max,i} - \sum_j x_{j,i} \tag{5}$$

doi:10.20944/preprints202112.0141.v1

Where, $Ebuy_{i,grid}$ is the energy bought by consumer i from the grid, $L_{max,i}$ is the maximum demand of consumer i , $x_{j,i}$ is the energy bought by consumer i from producer j in LEM. So, using Equation (4) and Equation (5), the second terms of Equation (1) and Equation (3) respectively could be found out.

2.1.3 Formulation of objective function

In Equation (1) and Equation (3), the clearing price is the equilibrium price at which supply and demand become same. The bid of each agent influences the market price, and thus a strong inter-dependency exists between profits and costs of agents. The aim of optimization is to maximize the overall average profits of the power system and to provide optimal solutions that distribute the earnings among all agents. Thus, the objective function is formulated as Equation (6).

$$\max.\sum_{j}\left(\frac{P_j}{N_p}\right)-\sum_{i}\left(\frac{C_i}{N_c}\right) \tag{6}$$

Where profit $\left(P_j\right)$ and cost $\left(C_i\right)$ are conflicting objectives since all agents try to achieve the best results for their own.

2.2 Representation of solutions and fitness function

As per Equation (6), Bi-level optimization tries to maximize profit of all agents by executing optimal bidding of agents in the

LEM. Let, $K=\{1,2,.....N_k\}$ is a set which consists of all agents (producers and consumers). The aim is to find the best tuple $\left(q_k,p_k\right)\forall k\in K$ representing the quantity of active power to bid and optimal price respectively in the LEM for each agent. The bidding is done for all $t\in T$ periods of optimization. For day-ahead market $T=24$ periods are taken. So, the vector of optimization variables $\vec{x}$ is represented as $\vec{x}=\left(\left[q_{k,t}\right]\bigcup\left[p_{k,t}\right]\right)$ where $q_{k,t}$ is the quantity and $p_{k,t}$ is the price the k^{th} agent will submit to the LEM at time period t. A sign convention is used to define consumer and producer type. A positive quantity represents a bid (i.e., buying in the LEM), and a negative quantity represents an offer (i.e., selling in the LEM). Thus, the action of an agent can be controlled by defining the bounds of variables in which consumer agents can submit bids in the LEM within the bounds $\left[0,L_{\max}\right]$ (i.e., between 0 and their maximum consumption limit), while producer agents can submit their offers within the bounds $\left[-PC_{\max,}0\right]$ (i.e., between 0 and their maximum power production capacity). The minimum and maximum boundary values for prices are in the range $\left[c^F,c^G\right]$ and same for all agents. Figure. 3 shows the structure of population of particles (solutions). Such particles are evaluated in the fitness function that returns the mean average profit of all agents plus the standard deviation as per Equation (7).

$$Fitness\left(\vec{x}\right)=\text{-}mean\left(Profits\right)+std(Profits) \tag{7}$$

Where $mean\left(profits\right)$ and $std\left(profits\right)$ are functions that compute the average and standard deviation of the profits that all agents obtained considering the bids/offers encoded in the particle. The negative sign in the first term is used to transform the profit maximization problem into a minimization problem. The less the value in Equation (7), the better the mean profits achieved by all agents. Equation (7) is optimized using the proposed CE-CMAES method, which is described, in the following section.

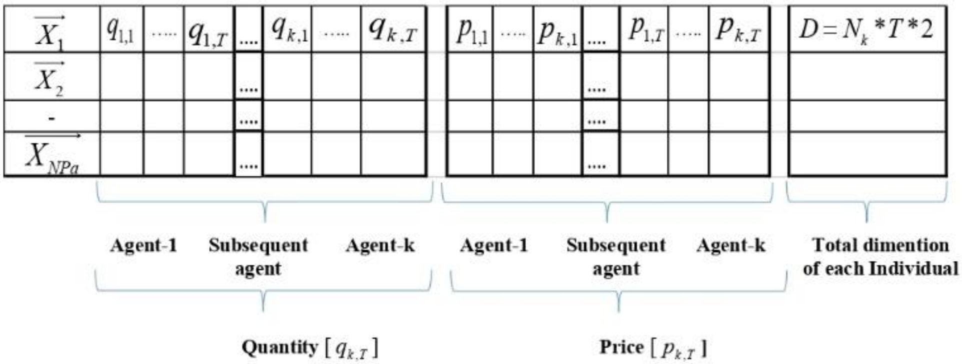

$\overrightarrow{X_1}$
$\overrightarrow{X_2}$
-
$\overrightarrow{X_{NPa}}$
$q_{1,1}$ ..... $q_{1,T}$ .... $q_{k,1}$ ..... $q_{k,T}$
$p_{1,1}$ ..... $p_{k,1}$ .... $p_{1,T}$ ..... $p_{k,T}$
$D = N_k * T * 2$
Agent-1
Subsequent agent
Agent-k
Agent-1
Subsequent agent
Agent-k
Total dimention of each Individual
Quantity [$q_{k,T}$]
Price [$p_{k,T}$]

Figure. 3. Presentation of an individual particle

3. CE-CMAES Method

CE-CMAES is a sequential hybridization of Cross Entropy (CE) proposed by Rubinstein [16] and CMAES was proposed by Nikolaus Hansen [17]. It has been applied to solve dynamic optimal power flow [18]. Both are population-based iterative meta-heuristic optimization methods similar to Particle Swarm Optimization [19] and Differential Evolutionary methods [20]. However, unlike conventional meta-heuristic methods which directly act upon prospective solutions (particles) to obtain optimal solutions, these methods generate "distribution" of prospective solutions and update it by updating the parameters of the distribution to obtain the optimal solutions. CE method was used for global exploration of the search space in the initial part (50% iterations). As it has small amount of parameters to be tuned, it can be easily implemented. CMAES method was used for local exploitation of search space for the remaining 50% iterations. It updates the mean of population such that the likelihood of elite particles near the sub-optimal solutions is increased. Covariance matrix is updated such that the likelihood of good points with better fitness and search steps is increased. Also, the evolution path provides a unique adaptive step size which prevents its premature convergence and thus enhances its local exploitation capability. The following section explains the theory of CE method.

3.1 Cross Entropy (CE) Method

The bids of all agents are optimized to minimize Equation (7). Firstly, a set of particles are randomly generated which obey the probability distribution function (pdf) $f(.;\theta)$, where θ is the vector of parameters to be optimized. Generally, $f(.;\theta)$ is the Gaussian distribution parameterized by its mean m and variance σ^2. i.e. $\theta = (m, \sigma^2)$. Secondly, a threshold value (δ) of the fitness function is selected and only those particles are considered whose fitness values are less than threshold value. i.e., $f(x) < \delta$. Such particles are known as "elite particles" (μ). Then, the new parameterized distribution function $f(.;\theta^n)$ with elite particles is updated to coincide the target distribution function $f(.;\theta^*)$ by minimizing Kullback-Leibler divergence. This procedure completes one iteration. In the subsequent generations, a family of distribution functions $f(.;\theta^{(1)})...f(.;\theta^*)$ are produced in accordance with $\delta^{(1)}...\delta^{(*)}$ to closely reach the optimal distribution function $f(.;\theta^*)$. The following section shows the step-by-step procedure of CE-CMAES.

Step: 1 Initialization of mean and standard deviation

Take generation $(g)=0$,Total no. of generations $g_{max}=500$, No. of particles $\left(NP_a\right)$, Dimension of the problem $\left(D\right)$, elite particles$\left(\mu\right)$ $\left(20\%-50\%\ of\ NP_a\right)$,smoothing parameters α and β, $m^{(g)}\in\mathbb{R}^D$ is the mean value of the search distribution for each dimension at generation $g,\sigma^{(g)}\in\mathbb{R}^+$ is standard deviation at generation g. x_{min} and x_{max} are the minimum and maximum boundary limits of D^{th} dimension particle. (i.e. $\left[0,L_{max}\right]$ for consumer bid quantities, $\left[-PC_{max,}\ 0\right]$ for producer offers quantities and $\left[c^F,c^G\right]$ for bid prices.) Mean and standard deviation of population are initialized as per Equation (8) and Equation (9) respectively.

$$m^{(0)}=\frac{\left(x_{min}+x_{max}\right)}{2} \tag{8}$$

$$\sigma^{(0)}=\frac{\left(x_{max}-x_{min}\right)}{4} \tag{9}$$

The conventional method to generate $\sigma^{(0)}$ is not applicable here because the values of decision variables are not known. So Equation (9) is used to initialize $\sigma^{(0)}$ as the maximum and minimum boundary values of decision variables are known.

Step: 2 Generation of population of particles

Generate the population of particles from the sampling distribution $N\left(m^{(g)},\sigma^{2(g)}\right)$ as per Equation (10).

$$x_i^{(g+1)}=m^{(g)}+\sigma^{(g)}randn()\quad for\ i=1,.....,NP_a \tag{10}$$

Where, NP_a is the total no. of particles (Population Size), $x_i^{(g+1)}\in\mathbb{R}^D$ is the ith particle obtained at generation $\left(g+1\right)$, $m^{(g)}\in\mathbb{R}^D$ is the mean value of the search distribution for each dimension at generation g, $\sigma^{(g)}\in\mathbb{R}^+$ is Step-size(Standard Deviation) at generation g , $randn()$ is a normally distributed random variable with parameters $N\left(0,1\right)$.

Step: 3 Adjustment of search distribution limit violations

The maximum and minimum boundary limits of all particles are checked and set as per Equation (11).

$$\begin{aligned} x_i^{(g+1)} &= x_i^{(g+1)},\ if\ x_{min}<x_i^{(g+1)}<x_{max} \\ x_i^{(g+1)} &= x_{min}, if\ x_i^{(g+1)}\le x_{min} \\ x_i^{(g+1)} &= x_{max}, if\ x_i^{(g+1)}\ge x_{max} \end{aligned} \tag{11}$$

Step: 4 Execution of Lower-level problem

Particles' bids and offers are used to determine market clearing price$\left(cp\right)$through merit order mechanism. Obtain energy transacted between the agents using Equation (4) and Equation (5).

Step: 5 Execution of Upper-Level Problem

Calculate the profit/cost of all agents using Equation (1) and Equation (3).

Step: 6 Fitness Function Evaluations and ranking

The fitness of all particles are evaluated by using Equation (7) which calculate the mean average profit of all agents and its standard deviation. Sort (rank) all fitness values in ascending order as given in Equation (12).

$$f\left(x_1\right)<f\left(x_2\right)<...<f\left(NP_a\right) \tag{12}$$

Where, $f(x_i)$ is the fitness of i^{th} particle, x_1 is a Global Best$\left(G_{Best}\right)$particle having the minimum fitness among all particles. Consider top best 20% particles as "elite" particles.

Step: 7 Updating the mean and standard deviation of elite particles

The mean $\left(m_{\mu}^{(g+1)}\right)$ of the selected elite particles is found by Equation (13).

$$m_{\mu}^{(g+1)}=\frac{1}{\mu}\sum_{i=1}^{\mu}x_{i:NP_a}^{(g+1)} \tag{13}$$

Where, $x_{i:NP_a}^{(g+1)}$ is the i^{th} best particle among whole population at $(g+1)$ iteration. The index $i:NP_a$ denotes the index of the i^{th} rank particle. Standard deviation $\left(\sigma_{\mu}^{(g+1)}\right)$ of elite particles is found by Equation (14).

$$\sigma_{\mu}^{(g+1)}=\sqrt{\frac{1}{\mu-1}\sum_{i=1}^{\mu}\left(x_{i:NP_a}^{(g+1)}-m^{(g+1)}\right)^2} \tag{14}$$

Step: 8 Smoothing of mean and standard deviation of population

As elite particles are near to sub-optimal solutions, more smoothing (weightage) is applied to their mean value as compared to mean of whole population as per Equation (15).

$$m^{(g+1)}=\alpha m_{\mu}^{(g+1)}+(1-\alpha)m^{(g+1)} \tag{15}$$

Where, Smoothing parameters $\alpha=0.9,\beta=0.1$:

Similarly, the standard deviation of elite particles should be modified to a small extent as they lie in the vicinity of sub-optimal solutions. So, less smoothing is applied to them as compared to standard deviation of all particles which require more exploration and thus more smoothing as given in Equation (16).

$$\sigma^{(g+1)}=\beta\sigma_{\mu}^{(g+1)}+(1-\beta)\sigma^{(g+1)} \tag{16}$$

Step: 9 Increment of generation count

Set $g:=g+1$

Go to step #2 and repeat the steps 2-9 until 50% of g_{max} are completed. So, CE method is used for global exploration. Subsequently, CMAES method is applied for local exploitation, which is discussed in the following section.

3.2 CMAES method

Step: 10 Initialization of mean of each dimension

Global Best (G_{Best}) particle obtained from CE method is considered as the mean value of corresponding dimension of CMAES as shown in Equation (17). As a result, the new search distributions (individuals) will be generated near to the sub-optimal solutions. So, the local exploitation of the search space becomes more effective as compared to with traditionally initialized mean value of each dimension.

$$m^g \leftarrow G_{best} \tag{17}$$

Covariance matrix C is initialized to Identity matrix with size $(D*D)$.

Step: 11 Sampling and Generation of population of particles

A population of particles (solutions or individuals) is generated by sampling a multivariate normal distribution as shown in Equation (18).

$$x_i^{(g+1)}=m^{(g)}+\sigma^{(g)}N\left(0,C^{(g)}\right) \quad for\ i=1,\dots..,NP_a \tag{18}$$

Where, NP_a is the total no. of particles (Population Size), $x_i^{(g+1)}\in\mathbb{R}^D$ is the ith particle obtained at generation $(g+1)$, $m^{(g)}\in\mathbb{R}^D$ is the mean value of the search distribution for each dimension at generation g, $\sigma^{(g)}\in\mathbb{R}^+$ is Step-size(Standard Deviation) at generation g, $C^{(g)}\in\mathbb{R}^{(D*D)}$ is the covariance matrix at generation g.

Step: 12 Adjustment of search distribution limit violations

The maximum and minimum boundary limits of all particles are checked and set as per Equation (11).

Step: 13 Execution of Lower-level problem

Particles' bids and offers are used to determine market clearing price (cp) through merit order mechanism. Obtain energy transacted between the agents using Equation (4) and Equation (5).

Step: 14 Execution of Upper-Level Problem

Calculate the profit/cost of all agents using Equation (1) and Equation (3)

Step: 15 Fitness Function Evaluations and ranking

The fitness of all particles are evaluated by using Equation (7) which calculate the profits/costs of all agents and the standard deviation between the profits/costs of agents. Sort (rank) all fitness values in ascending order and find G_{Best} and top 50% elite particles (μ) which have better fitness than remaining 50% particles.

CMAES is an iterative method in which $m^{(g+1)}, C^{(g+1)}$ and $\sigma^{(g+1)}$ are updated in every generation to obtain better solutions.

Step:16 Updating the Mean $\left(m^{(g+1)}\right)$

The updated mean of the search distribution is a weighted average of selected μ elite particles from the whole population as per Equation (19).

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i x_{i:NP_a}^{(g+1)} \tag{19}$$

$$\sum_{i=1}^{\mu} w_i = 1$$

$$w_1 \geq w_2 \geq ..... \geq w_\mu > 0$$

Where, $x_{i:NP_a}^{(g+1)}$ is the i^{th} best particle among whole population at $(g+1)$ generation. The index $i:NP_a$ denotes the index of the i^{th} rank particle. w_1 is the weightage applied to G_{Best} particle. As the highest weightage is applied to G_{Best} particle, the new individuals will be pulled towards it. Thus, update of mean maximizes the likelihood of elite particles near the sub-optimal solutions.

Step:17 Updating the Covariance Matrix $\left(C^{(g+1)}\right)$

A Covariance Matrix represents pairwise dependencies between the decision variables and it determines the shape of the distribution ellipsoid. It is updated such that the likelihood of good points with better fitness and search steps is increased. A covariance matrix of next generation is produced from the current covariance matrix as per Equation (20).

$$C^{(g+1)} = \left(1 - C_\mu \sum w_i\right) C^{(g)} + C_\mu \sum_{i=1}^{NP_a} w_i y_{i:NP_a}^{(g+1)} y_{i:NP_a}^{(g+1)^T} \tag{20}$$

where, C_μ is a learning rate for updating the covariance matrix, $0 \leq C_\mu \leq 1$

if $C_\mu = 0$: No learning takes place, if $C_\mu = 1$: No prior information is retained

$$w_1 \geq ... \geq w_\mu > 0 \geq w_{\mu+1} \geq w_{NP_a}$$

$$\sum_{i=1}^{\mu} w_i = 1$$

$$\sum_{i=1}^{NP_a} w_i \approx 0$$

$$y_{i:NP_a}^{(g+1)} = \frac{\left(x_{i:NP_a}^{(g+1)} - m^{(g)}\right)}{\sigma^{(g)}}$$

To enhance this step-size, the "evolution path" is also used. It is the sum of consecutive steps. To construct the evolution path $p_c^{(g+1)}$, the exponential smoothing is applied to assign more weightage to recent generations as per Equation (21).

$$p_c^{(g+1)} = (1-c_c)\, p_c^{(g)} + \left(\sqrt{c_c\,(2-c_c)\,\mu_{eff}}\right)\left(\frac{m^{(g+1)} - m^{(g)}}{\sigma^{(g)}}\right) \tag{21}$$

Where, evolution path at generation g is $p_c^{(g)} \in R^D$ and $p_c^{(0)} = 0$.

$c_c \leq 1$ is the learning rate.

$\left(\sqrt{c_c\,(2-c_c)\,\mu_{eff}}\right)$ is a normalization constant for p_c. $\mu_{eff\,=}\left(\sum_{i=1}^{\mu} w_i^2\right)^{(-1)}$ is variance effective selection mass $1 \leq \mu_{eff} \leq \mu$.

By using Equation (21), Rank-one update of the covariance matrix $C^{(g)}$ is obtained as per Equation (22).

$$C^{(g+1)} = (1-c_1)\, C^{(g)} + c_1 p_c^{(g+1)} {p_c^{(g+1)}}^T \tag{22}$$

Where, $c_1 \approx \dfrac{2}{D^2}$ is the learning rate.

The combination of (20) and (22) yields covariance matrix adaption equation as given in Equation (23).

$$C^{(g+1)} = \left(1-c_1-c_\mu \sum w_j\right) C^{(g)} + c_1 p_c^{(g+1)} {p_c^{(g+1)}}^T + c_\mu \sum_{i=1}^{NP_a} w_i y_{i:NP_a}^{(g+1)} \left(y_{i:NP_a}^{(g+1)}\right)^T \tag{23}$$

Where, $c_\mu \approx \min\left(\dfrac{\mu_{eff}}{n^2}, 1-c_1\right)$

Step: 18 Updating the Adaptive Step-size $\left(\sigma^{(g+1)}\right)$

The Step-size is controlled by an evolution path, which is a sum of successive steps independent and uncorrelated. The length of evolution path is decided by comparing its length with its expected value. If selection biases the evolution path to be longer than expected, then step-size is increased and vice versa. Ideally, selection does not bias the length of the evolution path and the length equals its expected length. A conjugate evolution path $\left(p_\sigma^{(g+1)}\right)$ is formed to get rid of dependency of the expected length of the evolution path on its direction as per Equation (24).

$$p_\sigma^{(g+1)} = (1-c_\sigma)\, p_\sigma^{(g)} + \sqrt{c_\sigma\,(2-c_\sigma)\,\mu_{eff}}\; C^{(g)^{-1/2}} \left(\frac{m^{(g+1)} - m^{(g)}}{\sigma^{g}}\right) \tag{24}$$

Where, $p_\sigma^{(g)} \in R^D$ is the conjugate evolution path at generation g with $p_\sigma^{(0)} = 0$.

$c_\sigma < 1$ is the learning rate for conjugate evolution path, $\sqrt{c_\sigma\,(2-c_\sigma)\,\mu_{eff}}$ is a normalization constant. $C^{(g)^{-1/2}}$ is calculated as per the Equation (25).

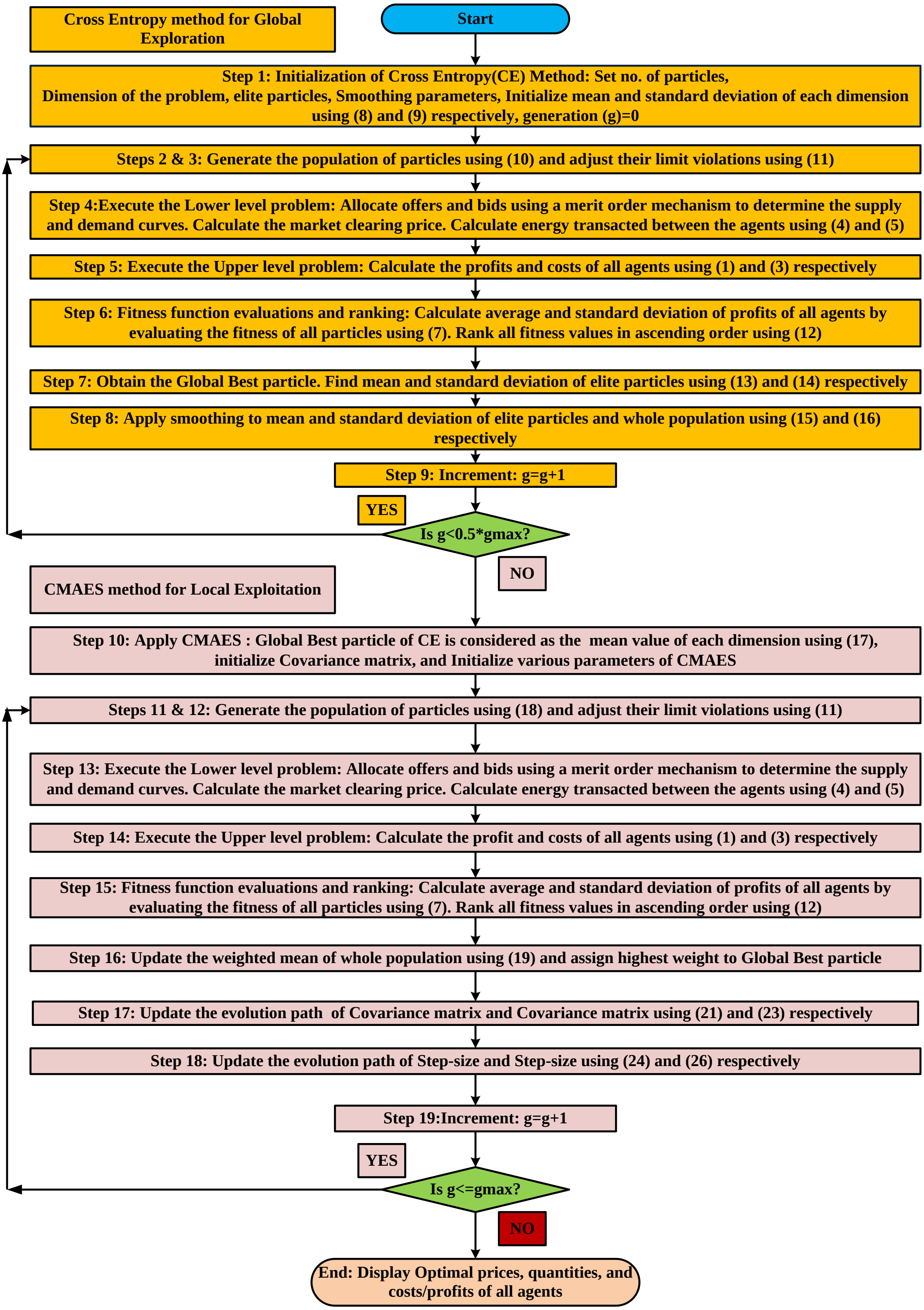

Cross Entropy method for Global Exploration
Start
Step 1: Initialization of Cross Entropy(CE) Method: Set no. of particles, Dimension of the problem, elite particles, Smoothing parameters, Initialize mean and standard deviation of each dimension using (8) and (9) respectively, generation (g)=0
Steps 2 & 3: Generate the population of particles using (10) and adjust their limit violations using (11)
Step 4:Execute the Lower level problem: Allocate offers and bids using a merit order mechanism to determine the supply and demand curves. Calculate the market clearing price. Calculate energy transacted between the agents using (4) and (5)
Step 5: Execute the Upper level problem: Calculate the profits and costs of all agents using (1) and (3) respectively
Step 6: Fitness function evaluations and ranking: Calculate average and standard deviation of profits of all agents by evaluating the fitness of all particles using (7). Rank all fitness values in ascending order using (12)
Step 7: Obtain the Global Best particle. Find mean and standard deviation of elite particles using (13) and (14) respectively
Step 8: Apply smoothing to mean and standard deviation of elite particles and whole population using (15) and (16) respectively
Step 9: Increment: g=g+1
YES
Is g<0.5*gmax?
NO
CMAES method for Local Exploitation
Step 10: Apply CMAES : Global Best particle of CE is considered as the mean value of each dimension using (17), initialize Covariance matrix, and Initialize various parameters of CMAES
Steps 11 & 12: Generate the population of particles using (18) and adjust their limit violations using (11)
Step 13: Execute the Lower level problem: Allocate offers and bids using a merit order mechanism to determine the supply and demand curves. Calculate the market clearing price. Calculate energy transacted between the agents using (4) and (5)
Step 14: Execute the Upper level problem: Calculate the profit and costs of all agents using (1) and (3) respectively
Step 15: Fitness function evaluations and ranking: Calculate average and standard deviation of profits of all agents by evaluating the fitness of all particles using (7). Rank all fitness values in ascending order using (12)
Step 16: Update the weighted mean of whole population using (19) and assign highest weight to Global Best particle
Step 17: Update the evolution path of Covariance matrix and Covariance matrix using (21) and (23) respectively
Step 18: Update the evolution path of Step-size and Step-size using (24) and (26) respectively
Step 19:Increment: g=g+1
YES
Is g<=gmax?
NO
End: Display Optimal prices, quantities, and costs/profits of all agents

Figure 4. Flowchart of CE-CMAES algorithm.

$$C^{(g)^{-1/2}} = B^{(g)} D^{(g)^{-1}} B^{(g)^{T}} \tag{25}$$

Where, $B^{(g)}$ is an orthonormal basis of eigenvectors and the diagonal elements of the diagonal matrix $D^{(g)}$ are square roots of the corresponding positive eigenvalues. The Euclidian length $\left(\left\|p_\sigma^{(g+1)}\right\|\right)$ of evolution path is compared with its expected length $\left(E\left\|N(0,I)\right\|\right)$ to update step size as given in Equation (26).

$$\sigma^{(g+1)} = \sigma^{(g)} \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\left\|p_\sigma^{(g+1)}\right\|}{E\left\|N(0,I)\right\|}-1\right)\right) \tag{26}$$

Where, $\sigma^{(g)} > 0$

$d_\sigma \approx 1$, is a damping parameter

$f\ \left\|p_\sigma^{(g+1)}\right\| > E\left\|N(0,I)\right\| \Rightarrow \sigma^{(g+1)}$ *increases*

$if\ \left\|p_\sigma^{(g+1)}\right\| < E\left\|N(0,I)\right\| \Rightarrow \sigma^{(g+1)}$ *decreases*

$if\ \left\|p_\sigma^{(g+1)}\right\| = E\left\|N(0,I)\right\| \Rightarrow \sigma^{(g+1)}$ *remains unchanged*

The step-size (standard deviation) is adaptively changed which ultimately prevents premature convergence of CMAES [21] and improves local exploitation of solution space.

Step: 19 Increment of generation count and Convergence check

Set $g := g+1$

Go to step #11 and execute steps 11-19 repeatedly until maximum number of generations are exhausted. The output of CE-CMAES are optimized prices, quantities (powers), and costs/profits of all agents. The flowchart of the Bi-level bidding optimization using CE-CMAES is given in Figure 4. The MATLAB™ codes of participated algorithms are available at http://www.gecad.isep.ipp.pt/ERM-competitions/2020-2/ [22].

4. Case Study and Result Analysis

In this section, the case study and the application of a set of algorithms to solve the bi-level optimal bidding problem in LEM are presented.

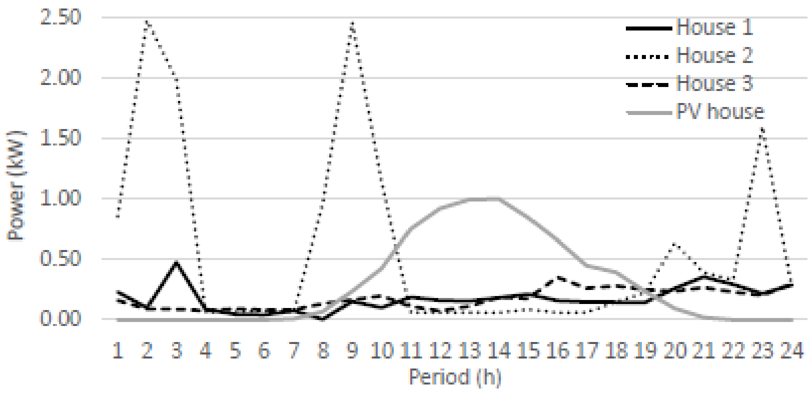


Figure 5. Profiles used in the case study. Ranges of power (in kW): house 1 [0.18-0.48], house 2 [0.06-2.50], house 3 [0.07- 0.36], PV (house) [0-1].

We use the case study from the competition that considers nine agents, 3 of which are consumers, 3 prosumers (i.e., consumers with PV generation capabilities), and 3 are small generators (i.e., CHP generators). The case study data generated from the reference power profiles of residential houses and PV systems were developed using the open datasets available on the PES ISS website [23]. Figure 5 shows the base profiles and range of power of three normal houses and a PV power profile used to generate the data. To do this, a randomized function with a uniform distribution around 20% of those profiles was applied. We also included generator agents represented as small CHPs generators with a maximum generation power of 2 kW and a marginal cost determined using Equation (2) with a factor of $b_{chp} = 0.18$ EUR/kWh [15]. At the end, feed-in and grid tariffs are set to $c^F = 0.12$ and $c^G = 0.28$ as in [15].

The purpose of the bi-level optimization is that all agents maximize their profits. With this purpose, the CE-CMAES algorithm was applied to solve the problem. To prove its effectiveness, the results of CE-CMAES were compared with 11 algorithms participating in the 2020 competition on "Evolutionary Computation in the Energy Domain: Smart Grid Applications" [22]. The 11 algorithms include Recursive Differential Grouping 3-Differential Evolutionary Particle Swarm Optimization (RDG3-DEEPSO), Ensembled method of CBBO, Cauchy and DEEPSO algorithm, Enhanced Hybrid Levy Particle Swarm Variable Neighborhood Search Optimization (EHL_PS_VNSO), CUMDANCauchy++: a Cellular EDA [24], HFEABC, Differential Evolutionary and Estimation of Distribution Algorithm (DEEDA), Differential Evolution and Teaching Learning Based Optimization (DE-TLBO), GASAPSO, AJSO, Hybrid Adaptive Differential Evolution with Decay Function (HyDE-DF) and Particle Swarm Optimization with Global Best Perturbation (PSO-GBP). To check the effectiveness and robustness of algorithms, we considered the 20 final solutions (one for each run) of the problem for each of the competing algorithms. A limit of 50,000 function evaluations were considered in each run. The results of all the above-mentioned algorithms were collected from the competition database [22].

4.1 Fine-Tuning of CE-CMAES Parameters

One of the most critical aspects of the design of any metaheuristic technique is the calibration of the strategic parameters. The CE-CMAES method has only two smoothing parameters α and β to be tuned. Thus, a variety of experiments had been carried out to evaluate their optimum values. In these experiments, we fixed the population size $NP_a = 100$, No. of Iteration (gmax)=500 and changed the smoothing parameters α and β within their feasible range of (0.7,1) and (0,0.3) respectively. The results of the experiments gave the optimum values of the smoothing parameters α and β, which were 0.9 and 0.1 respectively based on the best value of the mean fitness as shown in Table 1.

Table 1. Tuning the smoothing parameters of CE-CMAES

α	β	Mean Fitness
0.7	0.3	2.1128
0.75	0.25	2.1118
0.8	0.2	2.1112
0.85	0.15	2.1105
0.9	0.1	2.1088
0.95	0.05	2.1127
1	0	2.1184

Table 2. Fitness value of algorithms over 20

Run	FITNESS VALUE (EUR)											
	CE-CMAES	RDG3-DEEPSO	E-CBBO_Cauchy_DEEPSO	EHL_PS_VNSO	CUMDANCauchy++: a Cellular EDA	HFEABC	DEEDA	DE-TLBO	GASAPSO	AJSO	HyDE-DF	PSO-GBP
1	**2.113**	2.131	2.131	2.149	2.149	2.153	2.153	2.203	2.181	2.298	3.094	4.396
2	**2.109**	2.125	2.131	2.149	2.150	2.153	2.172	2.329	2.232	2.326	3.077	4.518
3	**2.105**	2.137	2.138	2.149	2.150	2.174	2.169	2.383	2.376	2.201	3.055	4.282
4	**2.103**	2.119	2.137	2.149	2.149	2.151	2.160	2.325	2.272	2.315	3.037	4.433
5	**2.106**	2.125	2.137	2.149	2.151	2.149	2.151	2.264	2.328	2.341	2.862	4.276
6	**2.110**	2.125	2.143	2.144	2.149	2.159	2.159	2.206	2.295	2.351	2.956	4.332
7	2.110	**2.107**	2.138	2.149	2.149	2.152	2.156	2.149	2.158	2.266	2.994	4.279
8	**2.110**	2.119	2.137	2.149	2.149	2.170	2.149	2.137	2.297	2.239	2.954	4.578
9	**2.105**	2.131	2.133	2.149	2.152	2.149	2.165	2.453	2.305	2.239	3.016	4.379
10	**2.117**	2.125	2.136	2.149	2.149	2.158	2.158	2.482	2.249	2.301	2.992	4.397
11	**2.098**	2.177	2.137	2.149	2.150	2.159	2.159	2.171	2.313	2.308	3.038	4.377
12	**2.103**	2.137	2.132	2.149	2.150	2.180	2.153	2.182	2.234	2.243	2.953	4.348
13	**2.108**	2.125	2.135	2.149	2.153	2.157	2.165	2.277	2.250	2.318	3.022	4.362
14	**2.111**	2.198	2.126	2.149	2.149	2.157	2.158	2.281	2.224	2.271	3.200	4.542
15	**2.103**	2.184	2.136	2.149	2.149	2.171	2.164	2.212	2.307	2.255	2.952	4.473
16	2.123	**2.119**	2.132	2.149	2.149	2.150	2.155	2.113	2.326	2.207	3.101	4.348
17	**2.119**	2.131	2.125	2.149	2.149	2.167	2.149	2.119	2.206	2.260	3.140	4.139
18	**2.105**	2.119	2.126	2.149	2.149	2.150	2.157	2.115	2.284	2.346	3.050	4.575
19	**2.101**	2.137	2.138	2.149	2.150	2.172	2.172	2.176	2.336	2.327	3.047	4.238
20	**2.108**	2.125	2.145	2.149	2.149	2.200	2.161	2.127	2.284	2.272	3.103	4.211

4.2 *Performance Comparison and Results*

After tuning the CE-CMAES smoothing parameters, the algorithm was applied to solve the bi-level optimization problem. To measure the CE-CMAES algorithm's performance, it was compared with contemporary state-of-the-art optimization algorithms in terms of fitness value for each run, as shown in Table 2. The fitness value displayed in Table 2 is the average value of the fitness function over the 50,000 function evaluations of each run. Table 2 clearly shows that CE-CMAES gets the best fitness value in eighteen runs out of 20 runs, where RDG3-DEEPSO have the best fitness value in two run out of 20 runs. It reveals the robustness of the CE-CMAES algorithm relative to other tested algorithms. Table 2 also reveals that the 11th run of CE-CMAES offers the best value of fitness of 2.0988 EUR out of 20 runs. It means that, in this run, CE-CMAES provides the best solution to the bi-level optimization problem. The optimal bids/offers and prices of all agents obtained by CE-CMAES in the eleventh run is given in Appendix A (Table A1).

All algorithms were run 20 times, and the results were recorded to obtain the mean value of those 20 runs. Table 3 gives the total overall costs of the system, the costs/profits of the group of agents (i.e., producers, prosumers and consumers), the mean fitness, the standard deviation, R.I. and the time taken by each algorithm to execute the 20 runs. The mean fitness is the average fitness over 20 runs (trials). The Ranking Index (R.I) is the sum of the mean fitness and the standard deviation over 20 runs. Table 3 also includes the overall costs/profits that agents can attain in the absence of a LEM (used as a baseline). The CE-CMAES algorithm achieved the first rank in terms of R.I with the lowest value of 2.1148. Also, EHL_PS_VNSO and CUMDANCauchy++: a Cellular EDA achieved the second and third rank with R.I of 2.1491 and 2.1506, respectively. Notice also that CE-CMAES achieves the best overall costs of 3.1840 EUR for the system compared to all tested algorithms. In terms of execution time, CE-CMAES takes an average of 17.0210 minutes to solve the problem, which is lower than the EHL_PS_VNSO (25.4820 minutes). It proves the advantage of hybridization of CMAES with CE, as the best solution obtained by CE is used for the initialization of CMAES. As a result, CMAES is near to the sub-optimal solutions. Therefore, CE-CMAES has good convergence capabilities. Although, some algorithms have less execution time as compared to CE-CMAES, but they have high value of R.I, mean fitness and overall costs. Overall, in terms of all the performance parameters, CE-CMAES gives the best solutions as compared to all tested algorithms.

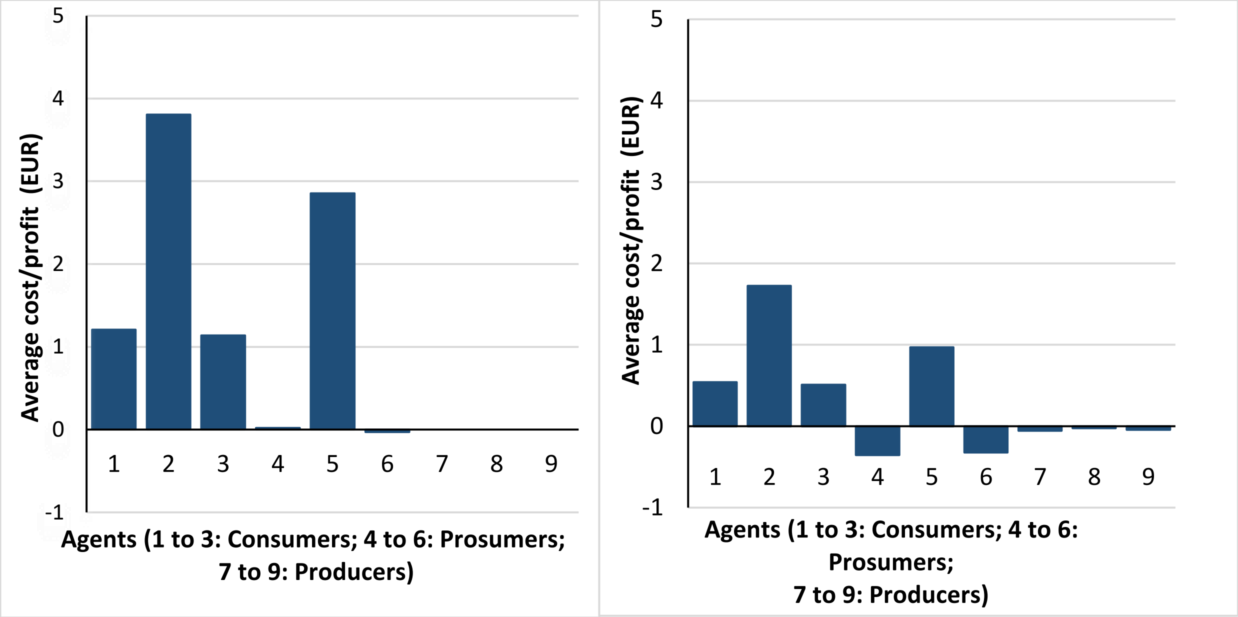
Table 3. Overall costs/profits, mean fitness, standard deviation, optimize time and R.I achieved by the tested algorithms.

Algorithms	Overall Costs (EUR)	Costs/profits by group of agents (EUR)			MeanFit (EUR)	Std	R.I	Time (mins)
		Costs		Profits				
		Consumers	Prosumers	Producers				
CE-CMAES	3.1840	2.7850	0.2950	0.1040	2.1088	0.0061	2.1148	17.0210
RDG3-DEEPSO	3.2528	2.8038	0.2978	0.1513	2.1353	0.0236	2.1589	16.5985
E- CBBO_Cauchy_DEEPSO	3.2762	2.8286	0.3191	0.1285	2.1536	0.0402	2.1938	26.2037
EHL_PS_VNSO	3.2732	2.7954	0.2889	0.1888	2.1446	0.0045	2.1491	25.4820
CUMDANCauchy++: a Cellular EDA	3.2867	2.7984	0.2896	0.1987	2.1500	0.0006	2.1506	13.8346
HFEABC	3.3081	2.8066	0.3018	0.1997	2.1584	0.0058	2.1642	1.8408
DEEDA	3.3125	2.8152	0.2974	0.1999	2.1605	0.0092	2.1697	16.9520
DE-TLBO	3.4716	2.9555	0.3957	0.1205	2.2357	0.1115	2.3473	16.0524
GASAPSO	3.5583	3.0624	0.4131	0.0828	2.2734	0.0554	2.3288	1.8695
AJSO	3.6664	3.0856	0.4606	0.1202	2.3130	0.0797	2.3927	3.6674
HyDE-DF	4.9823	3.9897	1.1648	-0.1722	3.0326	0.0773	3.1099	1.8578
PSO-GBP	7.8080	5.3994	2.1901	0.2185	4.3410	0.1493	4.4903	16.8510
No LEM	8.9985	6.1541	2.8444	0	4.9705	0	4.9705	0

In Table 3, relative to the baseline (a last row with No LEM), all groups of agents enhance their costs/profits. In the baseline, producer agents present '0' profits due to their unwillingness to sell their energy capacity to the grid at feed-in tariffs. The reason is the CHP generators have a marginal cost higher than the feed-in tariff. As a result, generators would not sell the energy to the grid if they could not recover at least their marginal costs.

doi:10.20944/preprints202112.0141.v1

To examine the profits made by independent agents, Figure 6 shows the profits made by individual agents. Figure 6(a) indicates the individual profit/cost in the baseline scenario, which confirms that, when no LEM is available, the safest choice for generators agents is not selling anything to the grid. That fact results in zero profits/costs for them, which is not suitable for the system as a whole. On the contrary, Figure 6(b) demonstrates that the best solution found by CE-CMAES improves the condition of agents, i.e., consumer agents 1 to 3 reduce their costs; prosumer agents 4 and 6 raise their profits and agent 5 reduces its cost; and producer agents 7 to 9 increase their profits.



(a) (b)

Figure 6. Overall profits achieved by individual agents. (a). Baseline solution. (b). Best solution found by CE-CMAES algorithm.

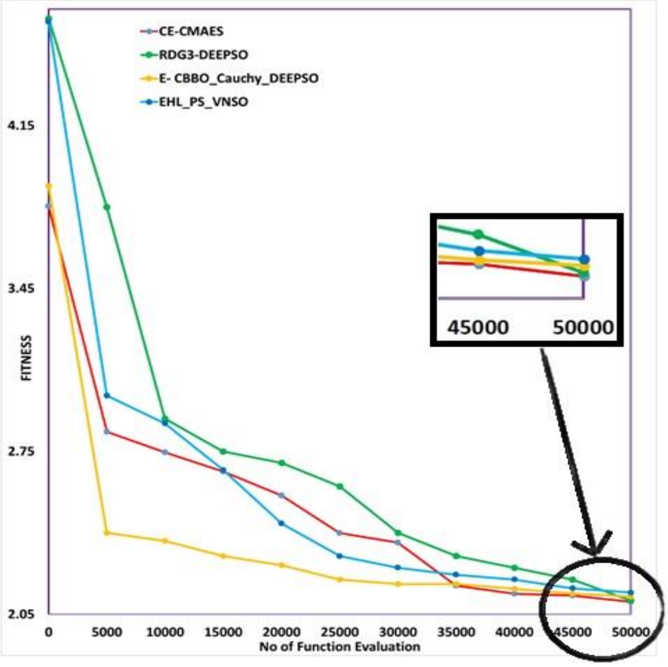


Figure 7. Convergence of fitness over number of function evaluation.

Finally, the convergence behaviour of the top four algorithms in terms of their fitness values have been analyzed. Figure 7 illustrates the fitness of the top four algorithms over 50,000 function evaluations. In the Figure 7, it can be seen that CE-CMAES gives slow convergence at start, but in final iterations convergence becomes fast and it achieves a better final solution compared to all evaluated algorithms. The CE-CMAES gives the slow convergence due to the 50-50 % contribution of CE and CMAES algorithms.

Table 4. Mean Fitness of CE-CMAES algorithm for different percentage usage of iteration in CE and CMAES.

CE	CMAES	Mean Fitness
10%	90%	2.8096
20%	80%	2.7466
30%	70%	2.7546
40%	60%	2.1107
50%	50%	2.1088
60%	40%	2.1132
70%	30%	2.119
80%	20%	2.1823
90%	10%	2.3354

4.3 Percentage Contribution of CE and CMAES in CE-CMAES

Table 4 shows, the mean fitness of CE-CMAES algorithm for different percentage usage of iterations in CE and CMAES out of total number of iterations. The comparison given in Table 4 proves that, CE-CMAES algorithm gives the best mean fitness, when both CE and CMAES are used for 50% of total iterations. This combination gives the slow convergence of CE-CMAES due to the time taken by the CMAES for the local exploitation of the search space, but finally it improves the solution in the last iterations as shown in Figure 7. Due to this reason, we used CE first for 50% of total iterations for the global exploration of search space and CMAES for remaining 50% of total iterations for the local exploitation of search space.

However, a comparison focused purely on mean fitness, R.I., and overall cost indicates an insufficient way of comparing results. Apart from the fact that CE-CMAES outperforms all the methods in terms of above all performance parameters, it is essential to check whether all the algorithms have a statistically significant difference in fitness as given in Table 3. For that, a statistical Wilcoxon signed rank test was used [25].

4.4 Wilcoxon Signed Rank Test

The Wilcoxon signed rank test is a non-parametric test that compares two paired groups (algorithms) to detect the significant differences between their behaviors. This test measures the difference in fitness between the pairs and analyses the differences. This test was carried out in a pairwise comparison between the CE-CMAES and the rest of contestant algorithms. The Wilcoxon signed rank test is used to accept (or reject) the null hypothesis that two samples represent two separate populations [26]. In this regard, a validation of the hypothesis confirms the performance of the algorithm. In this test, the significance level is set to 5% to verify all the tested algorithms' statistical differences. The Wilcoxon signed ranks test gives a meaningful finding, if the value of `P' given by all pairwise comparison is below 0.05, and then it may be assumed that there is statistical proof that any one or more algorithms are 95% substantially different from the other in terms of fitness value.

Table 5. Wilcoxon Signed Ranks Test Results.

PAIR	P-Value	T+	T-	Winner
CE-CMAES - E- CBBO_Cauchy_DEEPSO	0	0	210	+
CE-CMAES - EHL_PS_VNSO	0	0	210	+

CE-CMAES - CUMDANCauchy++: a Cellular EDA	0	0	210	+
CE-CMAES - HFEABC	0	0	210	+
CE-CMAES - DEEDA	0	0	210	+
CE-CMAES - DE-TLBO	0.006	0	210	+
CE-CMAES - GASAPSO	0	0	210	+
CE-CMAES - AJSO	0	0	210	+
CE-CMAES - HyDE-DF	0	0	210	+
CE-CMAES - PSO-GBP	0	0	210	+

To determine whether algorithm CE-CMAES reached a statistically better solution than other tested algorithms, or if not, whether the alternative hypothesis is valid, the sizes of the ranks provided by the Wilcoxon Signed-Rank Test (i.e., T+ and T-, as defined in [25] are examined. In Table 5, '+' indicates cases in which the null hypothesis is rejected and the CE-CMAES algorithm exhibited a statistically superior performance in the pair-wise Wilcoxon Signed-Rank Test at the 95% significance level (P=0.05). As can be seen in Table 5, the Wilcoxon signed ranks test gives the value of 'P' below 0.05 for all the pairwise treatment. This implies that, CE-CMAES outperforms the competition participated algorithms in each run. In terms of R.I, mean fitness, execution time and overall cost the CE-CMAES algorithm outperforms all tested algorithms. So overall, the CE-CMAES provides the best solutions for bi-level Optimization problem.

5. Conclusions and Future Works

This paper has presented the application of hybrid CE-CMAES method to solve complex bi-level bidding optimization problem in the context of local energy markets. The bi-level bidding optimization is difficult to solve since all agents intent to maximize their own profits at upper level, while modifying the market-clearing price in the local market lower level as a product of their bids. Thus, it creates a strong interdependence between both levels. It was demonstrated that CE-CMAES provides the best solutions in terms of mean fitness, Ranking Index (R.I) and overall cost and profit/cost of all agents compared with the state-of-the-art algorithms that participated in the competition "Evolutionary Computation in the Energy Domain: Smart Grid Applications". In addition, Wilcoxon Signed Rank Statistical test was used to prove that CE-CMAES is statistically different from the rest of tested algorithms. For future work, the CE-CMAES could be applied to the LEM bidding problem considering the effects that such process has in the distribution system (and also considering network constraints). In addition, the LEM model could be modified to include bidding optimization in external markets as well (e.g., wholesale market, capacity markets, and ancillary services markets). Finally, new hybridized methods based in the CE-CMAES approached could be analyzed to enhance the quality of solutions.

Author Contributions: Conceptualization, D.D., K.P.; methodology, D.D., K.P.; software, validation, K.P., D.D., J.S., F.L. and Z.V.; writing—original draft preparation, D.D.,K.S.; and T.J.; writing—review and editing, D.D., K.S., J.S., F.L. and Z.V.; supervision, K.P., J.S.

and F.L. Dharmesh Dabhi and Kartik Pandya contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.

Acknowledgements

This research was supported in part by Fundo Europeu de Desenvolvimento Regional (FEDER) Funds through COMPETE Program; by the National Funds through Fundação para a Ciência e a Tecnologia (FCT) under Projects PTDC/EEI-EEE/28983/2017 (CENERGETIC), UIDB/00760/2020, UIDB/00760/2020 and Project CEECIND /02814/2017.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A.

The optimal solution of the CE-CMAES algorithm is provided in Table A1.

Table A1. Solution of CE-CMAES.

Power (kW)/Price bid of	Hours																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Consumer (1 to 3) power bid	0.20	0.12	0.53	0.11	0.04	0.05	0.08	0.00	0.13	0.12	0.18	0.13	0.18	0.16	0.19	0.16	0.17	0.12	0.16	0.29	0.42	0.32	0.22	0.26
	0.87	2.07	1.76	0.08	0.06	0.08	0.07	0.97	2.21	1.34	0.06	0.07	0.07	0.07	0.10	0.07	0.05	0.18	0.19	0.67	0.38	0.39	1.55	0.24
	0.16	0.08	0.08	0.08	0.09	0.08	0.07	0.13	0.18	0.20	0.11	0.07	0.11	0.19	0.19	0.29	0.31	0.23	0.25	0.22	0.26	0.21	0.24	0.26
Prosumer (4 to 6) power bid	0.23	0.10	0.52	0.09	0.04	0.04	0.06	-0.06	0.00	-0.43	-0.65	-0.16	-0.67	-0.92	-0.40	-0.51	0.00	-0.28	0.00	0.17	0.38	0.33	0.19	0.31
	0.78	2.88	1.93	0.06	0.05	0.10	0.05	0.77	2.09	0.92	-0.09	-0.55	-0.70	-0.74	-0.61	-0.52	-0.30	0.00	0.00	0.44	0.30	0.28	1.45	0.26
	0.19	0.09	0.10	0.09	0.09	0.07	0.07	0.09	0.00	-0.27	-0.77	-0.79	-0.73	-0.23	-0.51	-0.16	0.00	0.00	0.02	0.12	0.25	0.24	0.22	0.29
Producer (7 to 9) power bid	-2.00	-2.00	-2.00	-2.00	-1.69	-1.58	-1.10	-1.35	-2.00	-1.34	-1.64	-0.33	-1.39	-1.66	-1.08	-0.94	-2.00	-1.87	-2.00	-2.00	-2.00	-2.00	-2.00	-1.05
	-1.26	-2.00	-2.00	-0.35	-1.10	-1.68	-1.84	-2.00	-2.00	-1.93	-0.56	-1.39	-1.05	0.00	-0.98	-0.79	0.00	-2.00	0.00	-1.04	-1.02	-0.89	-2.00	-0.71
	-2.00	-2.00	-2.00	-0.81	-1.42	-1.57	-0.93	-0.84	-2.00	-2.00	-1.42	-0.56	0.00	-1.13	-0.80	-1.00	-0.17	-0.78	-0.99	-1.09	-1.29	-0.83	-1.72	-2.00
Consumer (1 to 3) price bid	0.28	0.28	0.28	0.28	0.28	0.19	0.28	0.23	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
	0.25	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.14	0.28	0.28	0.28	0.28	0.28	0.28	0.28
	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
Prosumer (4 to 6) price bid	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.12	0.12	0.28	0.19	0.28	0.12	0.28	0.12	0.13	0.28	0.21	0.28	0.28	0.28	0.28	0.28
	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.12	0.27	0.12	0.12	0.12	0.12	0.28	0.28	0.18	0.28	0.28	0.22	0.28	0.26
	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.12	0.12	0.12	0.12	0.28	0.18	0.12	0.12	0.28	0.28	0.20	0.28	0.28	0.28	0.28	0.28
Producer (7 to 9) price bid	0.16	0.26	0.12	0.12	0.12	0.12	0.12	0.27	0.12	0.12	0.28	0.28	0.28	0.28	0.28	0.15	0.12	0.12	0.28	0.28	0.23	0.12	0.15	0.12
	0.12	0.28	0.18	0.20	0.12	0.12	0.12	0.12	0.26	0.12	0.12	0.12	0.28	0.12	0.12	0.28	0.12	0.21	0.12	0.27	0.12	0.14	0.28	0.28
	0.12	0.12	0.23	0.12	0.28	0.28	0.12	0.28	0.28	0.12	0.28	0.12	0.28	0.28	0.12	0.12	0.12	0.25	0.21	0.12	0.12	0.17	0.14	0.28

References

1. Soares J., Pinto T., Lezama F., and Morais H. Survey on complex optimization and simulation for the new power system paradigm. *Complexity* 2018; 1-32.
2. Xiao Y., Wang X., Pinson P. and Wang X.. A Local Energy Market for Electricity and Hydrogen. *IEEE Trans. Power Syst.* 2018; 33(4): 3898-3908.
3. Lezama F., Soares J., Hernandez-Leal P., Kaisers M., Pinto T., and Vale Z. Local Energy Markets: Paving the Path Toward Fully Transactive Energy Systems, *IEEE Trans. Power Syst.* 2019; 34(5): 4081-88.
4. Bakirtzis A., Ziogos N., Tellidou A., and Bakirtzis G. Electricity Producer Offering Strategies in Day-ahead Energy Market with Step-Wise Offers. *IEEE Trans. Power Syst.* 2007; 22(4): 1804-18.
5. Ghorani R., Fotuhi-Firuzabad M. and Moeini-Aghtaie M. Optimal Bidding Strategy of Transactive Agents in Local Energy Markets. *IEEE Trans. Smart Grid* 2019; 10(5):5152-62.
6. Rahimiyan M. and Baringo L. Strategic Bidding for a Virtual Power Plant in the Day-Ahead and Real-Time Markets: A Price-Taker Robust Optimization Approach. *IEEE Trans. Power Syst.* 2016; 31(4): 2676-87.
7. Liu G., Xu Y., and Tomsovic K. Bidding Strategy for Microgrid in Day-Ahead Market Based on Hybrid Stochastic/Robust Optimization. *IEEE Trans. Smart Grid* 2016; 07(01): 227-37.
8. Shafiee S., Zareipour H. and Knight A.M. Developing Bidding and Offering Curves of a Price-Maker Energy Storage Facility Based on Robust Optimization. *IEEE Trans. Smart Grid* 2019; 10(1): 650-60.
9. Jiang Y. et al. Optimal Bidding Strategy for a Power Producer under Monthly Pre-Listing Balancing Mechanism in Actual Sequential Energy Dual-Market in China. *IEEE Access*, 2019; 07:70986-998.

doi:10.20944/preprints202112.0141.v1

10. Wang Y., Wang D., Zhang H., and Zhang K. Optimal Bidding of Price-Maker Retailers with Demand Price Quota Curves Under Price Uncertainty. *IEEE Access* 2020; 08:120746-756.
11. Lezama F., Faia R., Soares J., Faria P. and Vale Z. Learning Bidding Strategies in Local Electricity Markets using Ant Colony optimization. in Proc.2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, United Kingdom, 2020; 1-8.
12. Clark P. A. and Westerberg A. W. Optimization for design problems having more than one objective. *Computers & Chemical Engineering* 1983; 07(4): 259-78.
13. Bard J. F. Some Properties of the Bilevel Programming Problem. *Journal of optimization theory and applications* 1991; 68(2): 371-78.
14. Lezama F., Soares J. and Vale Z. Optimal Bidding in Local Energy Markets using Evolutionary Computation. in Proc. 2019 20th International Conference on Intelligent System Application to Power Systems (ISAP), New Delhi, India, 2019; 1-6.
15. Mengelkamp E. M. Engineering local electricity markets for residential communities. Ph.D dissertation, Karlsruher Instituts fur Technologie (KIT), Karlsruhe, 2019.
16. Rubinstein R. Y. and Kroese D. P. The Cross-Entropy Method. A Unified Approach to Combinatorial Optimization. *Monte-Carlo Simulation, and Machine Learning,* New York: Springer; 2004.
17. Hansen N. The CMA Evolution Strategy: A Comparing Review, Berlin Heidelberg: Springer; 2006, 75-102.
18. Sarda J., Pandya K., and Lee K. Dynamic Optimal Power Flow with Cross Entropy Covariance Matrix Adaption Evolutionary Strategy for Systems with Electric Vehicles and Renewable Generators. *Int J Energy Res.* 2021; 45(7):10869-881.
19. Kennedy J. and Eberhart R. Particle swarm optimization. in Proc. of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995; 1942-48.
20. Storn R., and Price K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 1997; 11: 341–59.
21. Hansen N. The CMA Evolution Strategy: A Tutorial. Universite Paris-Saclay, 2016; URL [Online]. Available: https://arxiv.org/abs/1604.00772.
22. Soares J., Lezama F., Canizes B., and Vale Z. The Call for Competition on "Evolutionary Computation in the Energy Domain: Smart Grid Applications". 2019; [Online] Available: http://www.gecad.isep.ipp.pt/ERM-competitions/2020-2/.
23. Open data online at Website: http://sites.ieee.org/pes-iss/data-sets/.
24. Martínez-López Y., Rodríguez-González A. Y., Quintana J.M., Moya A., Morgado B., and Mayedo M. B. CUMDANCauchy-C1: a cellular EDA designed to solve the energy resource management problem under uncertainty. in Proceedings of the Genetic and Evolutionary Computation Conference Companion, Prague, Czech Republic, July 2019; 13–14.
25. Derrac J., Garca S., Molina D., and Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput*. 2011; 1(1): 3-18.
26. Garcia S., Fernandez A., Luengo J. and Herrera F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences* 2010; 180: 2044-64.